

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.:	10/669,931	§	Examiner:	Puente, Emerson C.
Filed:	September 24, 2003	§	Group/Art Unit:	2113
Inventor(s):		§	Atty. Dkt. No:	5760-13900/VRTS
	Hans F. van Rietschote, Mahesh	§		0394
	P. Saptarshi, and Craig W. Hobbs	§		
		§		
		§		
Title:	Providing High Availability	§		
	for an Application by	§		
	Rapidly Provisioning a	§		
	Node and Failing Over to	§		
	the Node	§		
		§		
		§		

APPEAL BRIEF

Mail Stop Appeal Brief - Patents

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed January 26, 2007, with PreAppeal Brief Request for Review, and the decision thereon mailed February 1, 2007, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The present application is owned by Veritas Operating Corporation, which is owned by Symantec Corporation. An assignment of the present application to the owner is recorded at Reel 014548, Frame 0188.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellant.

III. STATUS OF CLAIMS

Claims 1-8, 10-25, 27-34, and 36-41 are pending. Claims 9, 26, and 35 are cancelled. Claims 1-8, 10-15, 18-25, 27-34, and 36-41 are rejected under 35 U.S.C. § 103(a). Claims 16-17 are rejected under 35 U.S.C. § 102(e). It is these rejections that are being appealed. A copy of claims 1-8, 10-25, 27-34, and 36-41 is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMENTS

No unentered amendment to the claims has been filed after final rejection. The amendment filed December 18, 2006 was entered. See the Advisory Action mailed January 16, 2007.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method. The method comprises detecting that an application (14A) in a first node (10A) is to failover (decision block 50, Fig. 8) wherein the first node is included in a cluster (12A) being used to execute the application. The method further comprises adding a second node (10D) to the cluster responsive to the detecting (arrow 30, Fig. 2; block 56, Fig. 8). The method still further comprises provisioning the second node to execute the application responsive to the detecting

(arrow 30, Fig. 2; block 54, Fig. 8). The method still further comprises failing the application over from the first node to the second node (arrow 32, Fig. 3; block 60, Fig. 8). (See, e.g., Figs 1-3 and 8; and specification page 5, line 21-page 10, line 2; page 14, line 9-page 15, line 23).

Independent claim 16 is directed to a method. The method comprises detecting that an application (14A) in a first node (10A) is to failover (decision block 50, Fig. 8). The method further comprises provisioning a second node (10D) to execute the application responsive to the detecting (arrow 30, Fig. 2; block 54, Fig. 8). The method still further comprises attempting to failover the application from the first node to the second node (arrow 30, Fig. 3; block 60, Fig. 8); and detecting a lack of success in the failover, wherein the lack of success is due to a lack of an eligible node. Additionally, the method comprises permitting the application to execute on the first node responsive to the lack of the eligible node if the attempt to failover is due to a performance of the application on the first node being less than a threshold performance level. (See, e.g., Figs 1-3 and 8; and specification page 5, line 21-page 10, line 2; page 14, line 9-page 16, line 4).

Independent claim 19 is directed to a computer accessible medium (150, Fig. 10) encoded with instructions that, when executed: detect that an application (14A) in a first node (10A) is to failover (decision block 50, Fig. 8), wherein the first node is included in a cluster (12A) being used to execute the application; add a second node (10D) to the cluster responsive to detecting that the application is to failover (arrow 30, Fig. 2; block 56, Fig. 8); provision the second node to execute the application responsive to detecting that the application is to failover (arrow 30, Fig. 2; block 54, Fig. 8); and failover the application from the first node to the second node (arrow 32, Fig. 3; block 60, Fig. 8). (See, e.g., Figs 1-3 and 8; and specification page 5, line 21-page 10, line 2; page 14, line 9-page 15, line 23).

Independent claim 31 is directed to a system comprising a plurality of nodes (10A-10N). A first node (10B) of the plurality of nodes is configured to monitor

performance of an application (14A) executing on a second node (10A) of the plurality of nodes during use, and wherein, in response to a detection that the application is to failover from the first node, a third node (10D) is configured to be provisioned to execute the application, wherein the second node is included in a cluster (12A) being used to execute the application, and wherein the third node is added to the cluster responsive to the detection that the application is to failover from the second node during use, and wherein the application is failed over to the third node during use. (See, e.g., Figs 1-3 and 8; and specification page 5, line 21-page 10, line 2; page 14, line 9-page 15, line 23).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 16-17 are rejected under 35 U.S.C. § 102(e) as being anticipated by Harper et al., U.S. Patent No. 6,629,266 ("Harper").
2. Claims 1-4, 8, 10-13, 19-22, 25, 27-34, and 36-41 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vert et al., U.S. Patent No. 6,360,331 ("Vert") in view of Mashayekhi et al., U.S. Patent No. 6,922,791 ("Mashayekhi").
3. Claims 5-6, 14-15, 18, and 23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vert in view of Mashayekhi and Dinker et al., U.S. Patent No. 6,944,788 ("Dinker").
4. Claims 7 and 24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vert in view of Mashayekhi, Dinker, and Harper.

VII. ARGUMENT

First Ground of Rejection:

Claims 16-17 are rejected under 35 U.S.C. § 102(e) as being anticipated by Harper. Appellant traverses this rejection for at least the following reasons.

Claims 16-17:

Appellant respectfully submits that each of claims 16-17 recites a combination of features not taught or suggested in Harper. For example, claim 16 recites a combination of features including: "detecting that an application in a first node is to failover; [and] provisioning a second node to execute the application responsive to the detecting".

The Final Office Action mailed October 26, 2006 ("Office Action") asserts that Harper teaches the above highlighted features, citing col. 8, lines 11-16. However, these teachings are: "If the determination in step 502 is 'YES' (e.g., if the fail-to node can accept failover workload), then in step 505, the rejuvenation agent on the primary node instructs the cluster manager to gracefully (e.g., in a planned way) shut down the application on the primary node and in step 506 to restart the application on the secondary node." These teachings relate to failing over the node to an (already

provisioned) secondary node. Accordingly, these teachings have nothing to do with the provisioning features recited in claim 16.

Harper teaches that the primary node and the secondary (or backup) node are both configured to execute the application when the cluster is created. See, e.g., col. 6, lines 32-42: "Typically, in a two-node cluster, one node is designated the 'primary node' and normally runs the application software, and another is designated the 'backup node' and is capable of running the application when the primary node fails. Distributed cluster management software running on both the primary node and the secondary node continually checks on the health of the primary node and its associated application software." Therefore, Harper teaches that the backup/secondary node is provisioned to execute the application before execution begins on the primary node.

Nothing in Harper teaches or suggests "detecting that an application in a first node is to failover; [and] provisioning a second node to execute the application responsive to the detecting" as recited in claim 16. For at least the above stated reasons, Appellant submits that the rejection of claim 16 is in error and requests reversal of the rejection. The rejection of claim 17 (dependent from claim 16) is similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Claim 17 recites additional combinations of features not taught or suggested in the cited art.

Second Ground of Rejection:

Claims 1-4, 8, 10-13, 19-22, 25, 27-34, and 36-41 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vert in view of Mashayekhi. Appellant traverses this rejection for at least the following reasons.

Claims 1-3, 13, 19-21, 30-33, and 38-41:

Appellant respectfully submits that each of claims 1, 19, and 31 recite a combinations of features not taught or suggested in Vert and Mashayekhi. For example, claim 1 recites a combination of features including: "detecting that an application in a first node is to failover, wherein the first node is included in a cluster being used to execute the application; adding a second node to the cluster responsive to the detecting; [and] provisioning the second node to execute the application responsive to the detecting".

The Office Action alleges that Mashayekhi teaches the above highlighted features at col. 2, lines 60-67, asserting that the Examiner interprets "cluster" in the claims to be a cluster of active nodes as described in Mashayekhi. Appellant respectfully disagrees. The interpretation asserted by the Office Action is clearly contradicted by the plain language in Mashayekhi, in which the passive node is clearly part of the cluster and there is no cluster of active nodes the excludes the passive node. For example, Mashayekhi teaches: "Another known failover policy utilizes a separate 'passive' node that is present in the cluster exclusively for the purpose of being the failover node for all active nodes in the cluster. As illustrated in the following graph, each node on the cluster that is actively running applications (nodes 1-3) fails over to node 4, which is not tasked with running any applications other than in the event of a failover." (Mashayekhi, col. 2, lines 60-67). Thus, it is clear that Mashayekhi's cluster is four nodes, three of which are active and one of which is passive. All four nodes are clearly part of the cluster, and the passive node is provisioned *a priori* to execute any application from nodes 1 to 3 in the event of a failover. Thus, in the cited section, all that occurs when a failover event is detected is the act of failing over itself.

Accordingly, the cited section of Mashayekhi does not teach or suggest "detecting that an application in a first node is to failover, wherein the first node is included in a cluster being used to execute the application; adding a second node to the cluster responsive to the detecting; [and] provisioning the second node to execute the application responsive to the detecting " as recited in claim 1. Vert does not teach or suggest the above highlighted features, either. Accordingly, the alleged combination of Vert and Mashayekhi does not teach or suggest the combination of features recited in claim 1.

Claim 19 recites a combination of features including: "detect that an application in a first node is to failover...add a second node to the cluster responsive to detecting that the application is to failover; provision the second node to execute the application responsive to detecting that the application is to failover". The same teachings of Vert and Mashayekhi highlighted above with regard to claim 1 are alleged to teach the above highlighted features of claim 19. Appellant respectfully submits that Vert and Mashayekhi do not teach or suggest the above highlighted features, either. Accordingly, the rejection of claim 19 is also in error.

Claim 31 recites a combination of features including: "a third node is configured to be provisioned to execute the application...wherein the third node is added to the cluster responsive to the detection that the application is to failover from the second node during use". The same teachings of Vert and Mashayekhi highlighted above with regard to claim 1 are alleged to teach the above highlighted features of claim 31. Appellant respectfully submits that Vert and Mashayekhi do not teach or suggest the above highlighted features, either. Accordingly, the rejection of claim 31 is also in error.

For at least the above stated reasons, Appellant submits that the rejection of claims 1, 19, and 31 is in error and requests reversal of the rejection. The rejection of claims 2-3, 13, and 39 (dependent from claim 1); claims 20-21, 30, and 40 (dependent from claim 19); and claims 32-33, 38, and 41 (dependent from claim 31) are similarly in error for at least the above stated reasons, and reversal of the rejection is requested. Each of claims 2-3, 13, 20-21, 30, 32-33, and 38-41 recites additional combinations of features

not taught or suggested in the cited art.

Claims 4, 22, and 34:

Claims 4, 22, and 34 depend from claims 1, 19, and 31, respectively. Accordingly, the rejection of claims 4, 22, and 34 is in error for at least the reasons highlighted above with regard to claims 1, 19, and 31. Additionally, claim 4 recites a combination of features including: "the second node has multiple boot capability, and wherein the provisioning comprises rebooting the second node from a partition that comprises one or more resources used by the application."

The Office Action alleges that Vert teaches the above highlighted features, citing col. 9, lines 5-11. However, these teachings are: " For example, if a resource (e.g., an application) fails, the failover manager 87 may choose to restart the resource, or to take the resource offline along with any resources dependent thereon. If the failover manager 87 takes the resource offline, the group is restarted on another system in the cluster, known as pushing the group to another system. A cluster administrator may also manually initiate such a group transfer." These teachings describe details of either restarting a failing resource (application) on a node, or failing over the application from one node to another. This has nothing to do with a node that has multiple boot capability. Restarting a resource is not rebooting the node. In fact, there is nothing in these teachings about booting a node, nor rebooting. Restarting a service has nothing to do with rebooting the node as a whole. Accordingly, Vert fails to teach or suggest "the second node has multiple boot capability, and wherein the provisioning comprises rebooting the second node from a partition that comprises one or more resources used by the application."

Claim 22 recites a combination of features including: "the second node has multiple boot capability, and wherein the instructions which, when executed, provision the second node comprise instructions which, when executed, reboot the second node from a partition that comprises one or more resources used by the application". The same teachings of Vert highlighted above with regard to claim 4 are alleged to teach the

features of claim 22. Appellant respectfully submits that Vert does not teach or suggest the above features, either.

Claim 34 recites a combination of features including: "the third node has multiple boot capability, and wherein provisioning the third node comprises rebooting the third node from a partition that comprises one or more resources used by the application." The same teachings of Vert highlighted above with regard to claim 4 are alleged to teach the features of claim 34. Appellant respectfully submits that Vert does not teach or suggest the above features, either.

For at least the above stated reasons, Appellant submits that the rejection of claims 4, 22, and 34 is in error and requests reversal of the rejection.

Claims 8 and 25:

Claims 8 and 25 depend from claims 1 and 23, respectively. Accordingly, the rejection of claims 8 and 25 is in error for at least the reasons highlighted above with regard to claims 1 and 19. **Still further, the rejection of claim 25 is clearly in error since claim 23, on which claim 25 depends, is not rejected over the combination of Vert and Mashavekhi.**

Additionally, claim 8 recites a combination of features including: "adding the first node to the plurality of nodes to be selectable for provisioning." The Office Action alleges that Vert teaches these features, citing col. 4, line 63 to col. 5, line 5. However, these teachings are: "To create a new cluster, a system administrator runs a cluster installation utility on a system that then becomes a first member of the cluster 58. For a new cluster 58, a database is created and the initial cluster member information is added thereto. The administrator then configures any devices that are to be managed by the cluster software. At this time, a cluster exists having a single member, after which the installation procedure is run on each of the other members of the cluster. For each added member, the name of the existing cluster is entered and the new system receives a copy of the existing cluster database." These teachings describe how a cluster is created.

Claim 8 recites adding the first node (from which the application has failed over) to the nodes that can be provisioned and added to another cluster. The above teachings are silent on these features.

Claim 25 recites a combination of features including: "the instructions, when executed, add the first node to the plurality of nodes to be selectable for provisioning". The same teachings of Vert highlighted above with regard to claim 4 are alleged to teach the features of claim 25. Appellant respectfully submits that Vert does not teach or suggest the above features, either.

For at least the above stated reasons, Appellant submits that the rejection of claims 8 and 25 is in error and requests reversal of the rejection.

Claims 10, 12, 27, 29, and 36:

Claims 10, 27, and 36 depend from claims 1, 19, and 31, respectively. Accordingly, the rejection of claims 10, 27, and 36 is in error for at least the reasons highlighted above with regard to claims 1, 19, and 31. Additionally, claim 10 recites a combination of features including: "detecting that the performance of the application executing on the first node is less than a threshold performance level."

The Office Action alleges that Vert teaches the above highlighted features, asserting that Vert discloses sending periodic messages called heartbeats to detect that the communication path is good and that the other system is operational, citing Vert col. 5, lines 30-35. The Office Action goes on to assert that in the event of a communication failure (no heartbeat), the system fails over to one or more active systems. The Office Action concludes that the heartbeats represent the performance of the application, and that when heartbeats are not received the application is less than a threshold performance level (See Office Action, page 6, lines 8-14). Appellant respectfully disagrees.

The heartbeats are used to detect that a given system is still operating and in connection with other systems. The heartbeats provide no information regarding the

performance of the application on the system. The presence of a heartbeat does not indicate that the application is achieving good performance, indeed the performance may be poor even though the heartbeats are being provided on time. Similarly, the absence of the heartbeats does not indicate poor performance, but rather indicates a failure on a system or on the communication link. An application may be performing well, but the communication link may have failed, for example. Furthermore, there is no threshold described in the cited section. If Vert's heartbeat has stopped because of a system failure, then the application is no longer executing, and thus there is no detection of a performance level of an application executing on the node (as recited in claim 10).

Claim 27 recites a combination of features including: "the instructions which, when executed, detect that the application is to failover comprise instructions which, when executed, detect that the performance of the application executing on the first node is less than a threshold performance level". The same teachings of Vert highlighted above with regard to claim 10 are alleged to teach the features of claim 27. Appellant respectfully submits that Vert does not teach or suggest the above features, either.

Claim 36 recites a combination of features including: "the first node is configured to detect that the performance of the application executing on the second node is less than a threshold performance level". The same teachings of Vert highlighted above with regard to claim 10 are alleged to teach the features of claim 36. Appellant respectfully submits that Vert does not teach or suggest the above features, either.

For at least the above stated reasons, Appellant submits that the rejection of claims 10, 27, and 36 is in error and requests reversal of the rejection. The rejection of claim 12 (dependent from claim 10) and claim 29 (dependent from claim 27) is similarly in error for at least the above stated reasons. Each of claims 12 and 29 recites additional combinations of features not taught or suggested in the cited art.

Claims 11, 28, and 37:

Claims 11, 28, and 37 depend from claims 10, 27, and 36, respectively.

Accordingly, the rejection of claims 11, 28, and 27 is in error for at least the reasons highlighted above with regard to claims 10, 27, and 36. Additionally, each of claims 11, 28, and 37 recites a combination of features including: "the performance is less than the threshold performance level for at least a predefined time interval."

The Office Action continues to allege that the heartbeats described in Vert represent the performance level of the application. However, as noted above, the heartbeats do not correlate to the performance level. Thus, the heartbeats have nothing to do with the performance being less than a threshold level for at least a predefined time interval, either.

For at least the above stated reasons, Appellant submits that the rejection of claims 11, 28, and 37 is in error and requests reversal of the rejection.

Third Ground of Rejection:

Claims 5-6, 14-15, 18, and 23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vert in view of Mashayekhi and Dinker. Appellant traverses this rejection for at least the following reasons.

Claims 5, 14, and 23:

Claims 5, 14, and 23 depend from claims 1, 1, and 19, respectively. Accordingly, the rejection of claims 5, 14, and 23 is in error for at least the reasons highlighted above with regard to claims 1 and 19. The Office Action relies on Dinker to allegedly teach various features of claims 5, 14, and 23. However, Dinker does not teach the features of claims 1 and 19 highlighted above. Accordingly, the combination of Vert, Mashayekhi, and Dinker do not teach the features of claims 1 and 19, and thus cannot teach the combinations of features 5, 14, and 23 dependent therefrom.

For at least the above stated reasons, Appellant submits that the rejection of claims 5, 14, and 23 is in error and requests reversal of the rejection.

Claim 6:

Claim 6 depends from claim 5 and thus the rejection of claim 6 is in error for at least the reasons highlighted above with regard to claim 5. Furthermore, claim 6 recites a combination of features including: "the second node is executing a different application when selected [to be provisioned for failover of the application from the first node]".

The Office Action alleges that Mashayekhi teaches the above highlighted features, citing column 9, lines 22-27. However, this section describes a time based failover policy that appears to have nothing to do with the above highlighted features of claim 6. The Office Action asserts that Mashayekhi discloses that the cluster is not running any application prior to failover and running applications actively when there is a failover. Appellant respectfully disagrees...there is no teaching that the cluster is not running any application prior to failover. If Mashayekhi did have such a teaching there would be nothing to failover since there would be no application running. Finally, the

Office Action refers to col. 2, lines 65-67. However, this section teaches that nodes 1-3 (actively running applications) fail over to node 4 which is not tasked with running applications other than in the event of failover. Thus, node 4 is not executing a different application at the time the failover occurs.

For at least the above stated reasons, Appellant submits that the rejection of claim 6 is in error and requests reversal of the rejection.

Claim 15:

Claim 15 depends from claim 14 and thus the rejection of claim 15 is in error for at least the reasons highlighted above with regard to claim 14. Additionally, claim 15 recites a combination of features including: "provisioning a third node to execute the application responsive to detecting the lack of success; and failing the application over from the second node to the third node."

The Office Action asserts that Dinker teaches the above highlighted features, stating that Dinker discloses a primary application server that fails over to a backup server (which becomes the new primary server) and then the new primary server fails over to another backup server (see Office Action, page 12, lines 3-11). Appellant does not disagree that Dinker teaches failing over from the primary server to a backup server, which may then fail over to another server. However, Dinker lacks any teaching of provisioning the servers in response to detecting the failover/lack of success of a failover. Instead, Dinker (like Vert and Mashayekhi) only teaches failing over to servers that have been provisioned *a priori* to execute the application. Thus, the alleged combination of Vert, Mashayekhi, and Dinker fails to teach "provisioning a third node to execute the application responsive to detecting the lack of success; and failing the application over from the second node to the third node" as recited in claim 15.

For at least the above stated reasons, Appellant submits that the rejection of claim 15 is in error and requests reversal of the rejection.

Claim 18:

Claim 18 depends from claim 1 and thus the rejection of claim 18 is in error for at least the reasons highlighted above with regard to claim 1. Additionally, claim 18 recites a combination of features including: "determining that a performance level on the second node is less than a threshold; provisioning a third node to execute the application responsive to the determining; and failing the application over from the second node to the third node."

The Office Action asserts that Dinker teaches the above highlighted features, again relying on Dinker to disclose a primary application server that fails over to a backup server (which becomes the new primary server) and then the new primary server fails over to another backup server (see Office Action, page 12, lines 12-bottom, extending to page 13). Appellant does not disagree that Dinker teaches failing over from the primary server to a backup server, which may then fail over to another server. However, Dinker lacks any teaching of provisioning the servers in response to detecting that performance is below a threshold. Instead, Dinker (like Vert and Mashayekhi) only teach failing over to servers that have been provisioned *a priori* to execute the application. Thus, the alleged combination of Vert, Mashayekhi, and Dinker fails to teach "provisioning a third node to execute the application responsive to the determining [that a performance level on the second node is less than a threshold]".

Furthermore, Dinker does not teach determining that a performance level is less than a threshold. The Office Action relies on teachings of heartbeats in Dinker that are similar to Vert's heartbeats. For the same reasons discussed above with regard to Vert, Dinker's heartbeats are not correlated to the performance level of the application and thus do not teach determining that the performance level is less than a threshold.

For at least the above stated reasons, Appellant submits that the rejection of claim 18 is in error and requests reversal of the rejection.

Fourth Ground of Rejection:

Claims 7 and 24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Vert in view of Mashayekhi, Dinker, and Harper. Appellant traverses this rejection for at least the following reasons.

Claim 7 and 24:

Claims 7 and 24 depend from claims 5 and 23, respectively, and thus the rejection of claims 7 and 24 is in error for at least the reasons highlighted above with regard to claims 5 and 23. The Office Action relies on Harper to allegedly teach various features of claims 7 and 24. However, Harper does not teach the features of claims 5 and 23 (nor the features of claims 1 and 19, on which they depend). Accordingly, the combination of Vert, Mashayekhi, Dinker, and Harper do not teach the features of claims 1 and 19, and 5 and 23, and thus cannot teach the combinations of features of claims 7 and 24 dependent therefrom.

For at least the above stated reasons, Appellant submits that the rejection of claims 7 and 24 is in error and requests reversal of the rejection.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of claims 1-8, 10-25, 27-34, and 36-41 are erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of \$500 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-13900/LJM. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

/Lawrence J. Merkel/
Lawrence J. Merkel, Reg. #41,191
Agent for Appellant

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: March 20, 2007

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method comprising:

detecting that an application in a first node is to failover, wherein the first node is included in a cluster being used to execute the application;

adding a second node to the cluster responsive to the detecting;

provisioning the second node to execute the application responsive to the detecting; and

failing the application over from the first node to the second node.

2. The method as recited in claim 1 wherein the provisioning comprises activating one or more resources used by the application on the second node.

3. The method as recited in claim 1 wherein the provisioning comprises installing one or more resources used by the application on the second node.

4. The method as recited in claim 1 wherein the second node has multiple boot capability, and wherein the provisioning comprises rebooting the second node from a partition that comprises one or more resources used by the application.

5. The method as recited in claim 1 further comprising selecting the second node from a plurality of nodes.

6. The method as recited in claim 5 wherein the second node is executing a different application when selected.

7. The method as recited in claim 5 wherein the selecting comprises verifying that the second node includes hardware that is sufficient to execute the application.
8. The method as recited in claim 1 further comprising adding the first node to the plurality of nodes to be selectable for provisioning.
10. The method as recited in claim 1 wherein the detecting comprises detecting that the performance of the application executing on the first node is less than a threshold performance level.
11. The method as recited in claim 10 wherein the performance is less than the threshold performance level for at least a predefined time interval.
12. The method as recited in claim 10 wherein the detecting comprises alternatively detecting a failure in a service group including the application.
13. The method as recited in claim 1 wherein the detecting comprises detecting a failure in a service group including the application.
14. The method as recited in claim 1 further comprising detecting a lack of success in the failing over.
15. The method as recited in claim 14 further comprising:
- provisioning a third node to execute the application responsive to detecting the lack of success; and
- failing the application over from the second node to the third node.
16. A method comprising:

detecting that an application in a first node is to failover;

provisioning a second node to execute the application responsive to the detecting;

attempting to failover the application from the first node to the second node;

detecting a lack of success in the failover, wherein the lack of success is due to a lack of an eligible node; and

permitting the application to execute on the first node responsive to the lack of the eligible node if the attempt to failover is due to a performance of the application on the first node being less than a threshold performance level.

17. The method as recited in claim 16 wherein, if the attempt to failover is due to a failure in a service group including the application, the method further comprises notifying an administrator.

18. The method as recited in claim 1 further comprising:

determining that a performance level on the second node is less than a threshold;

provisioning a third node to execute the application responsive to the determining;

and

failing the application over from the second node to the third node.

19. A computer accessible medium encoded with instructions that, when executed:

detect that an application in a first node is to failover, wherein the first node is included in a cluster being used to execute the application;

add a second node to the cluster responsive to detecting that the application is to failover;

provision the second node to execute the application responsive to detecting that the application is to failover; and

failover the application from the first node to the second node.

20. The computer accessible medium as recited in claim 19 wherein the instructions which, when executed, provision the second node comprise instructions which, when executed, activate one or more resources used by the application on the second node.

21. The computer accessible medium as recited in claim 19 wherein the instructions which, when executed, provision the second node comprise instructions which, when executed, install one or more resources used by the application on the second node.

22. The computer accessible medium as recited in claim 19 wherein the second node has multiple boot capability, and wherein the instructions which, when executed, provision the second node comprise instructions which, when executed, reboot the second node from a partition that comprises one or more resources used by the application.

23. The computer accessible medium as recited in claim 19 wherein the instructions, when executed, select the second node from a plurality of nodes.

24. The computer accessible medium as recited in claim 23 wherein the instructions which, when executed, select the second node comprise instructions which, when executed, verify that the second node includes hardware that is sufficient to execute the application.

25. The computer accessible medium as recited in claim 23 wherein the instructions,

when executed, add the first node to the plurality of nodes to be selectable for provisioning.

27. The computer accessible medium as recited in claim 19 wherein the instructions which, when executed, detect that the application is to failover comprise instructions which, when executed, detect that the performance of the application executing on the first node is less than a threshold performance level.

28. The computer accessible medium as recited in claim 27 wherein the performance is less than the threshold performance level for at least a predefined time interval.

29. The computer accessible medium as recited in claim 27 wherein the instructions which, when executed, detect that the application is to failover comprise instruction which, when executed, alternatively detect a failure in a service group including the application.

30. The computer accessible medium as recited in claim 19 wherein the instructions which, when executed, detect that the application is to failover comprise instruction which, when executed, detect a failure in a service group including the application.

31. A system comprising a plurality of nodes, wherein a first node of the plurality of nodes is configured to monitor performance of an application executing on a second node of the plurality of nodes during use, and wherein, in response to a detection that the application is to failover from the first node, a third node is configured to be provisioned to execute the application, wherein the second node is included in a cluster being used to execute the application, and wherein the third node is added to the cluster responsive to the detection that the application is to failover from the second node during use, and wherein the application is failed over to the third node during use.

32. The system as recited in claim 31 wherein provisioning the third node comprises activating one or more resources used by the application on the second node.

33. The system as recited in claim 31 wherein provisioning the third node comprises installing one or more resources used by the application on the third node.
34. The system as recited in claim 31 wherein the third node has multiple boot capability, and wherein provisioning the third node comprises rebooting the third node from a partition that comprises one or more resources used by the application.
36. The system as recited in claim 31 wherein the first node is configured to detect that the performance of the application executing on the second node is less than a threshold performance level.
37. The system as recited in claim 36 wherein the performance is less than the threshold performance level for at least a predefined time interval.
38. The system as recited in claim 31 wherein the second node is configured to detect a failure in a service group including the application, and wherein the application is to failover from the second node if the second node detects the failure.
39. The method as recited in claim 1 further comprising removing the first node from the cluster responsive to successfully failing over the application to the second node.
40. The computer accessible medium as recited in claim 19 wherein the instructions, when executed, remove the first node from the cluster responsive to successfully failing over the application to the second node.
41. The system as recited in claim 31 wherein the second node is removed from the cluster responsive to a successful failover to the third node.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings known to Appellant.